[redacted]

Report No. 368 COO-1018-1199

GRAPH TRANSFORMATIONS FOR COMPOSITE FORMATION

by

John C. Schwebel

December 12, 1969

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

COO-1018-1195

Report No. 368

GRAPH TRANSFORMATIONS FOR COMPOSITE FORMATION

by

John C. Schwebel

December 12, 1969

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

*This work was supported by the U.S. Atomic Energy Commission under Contract No. USAEC AT(11-1)-1018.

1. INTRODUCTION

This paper defines and investigates transformations of labelled directed graphs. The properties of composite formation under a binary relation, which were defined in a previous report [1], are shown to be sufficient conditions for some graph transformations. Implications between the defined transformations are also determined. A method of classifying relations involved in a transformation is demonstrated. A partial classification is performed on the basis of the defined transformations and on the abstract properties of the constituent relations.

The motivation for the work in this area was the concept of "precedence" of graph transformations, introduced by Professor Bruce H. McCormick. These ideas were put forth at a computer graphics conference [2] and in further frequent discussions since that time with this author.

Within the framework of these ideas, an abstract graph is "parsed" by a series of composite forming transformations. In forming composites some links, representing relations between objects, must be assumed to have a higher binding strength, i.e. precedence, than others. Each transformation simplifies the available information by deleting or "cutting" links of lower precedence to objects on one level and extending them in turn to a composite element introduced at a higher level in the parse. Information supplied by relations within the composite element, of course, remains available at the lower level in the parse.

2. LABELLED GRAPHS AND TRANSFORMATIONS

We will consider finite directed graphs with labelled edges corresponding to binary relations.

A <u>labelled graph</u>, $L = (X, H_\Delta)$ consists of an arbitrary set, X, of n elements and a set , H_Δ, of m binary relations on X:

$$H_\Delta = \{H_\delta \mid H_\delta \subseteq X \times X,\ H_\delta \neq \emptyset,\ \delta \in \Delta\}$$

Here $\Delta = \{\delta\}$ is an index set of m elements. L is represented by a diagram with n points corresponding to elements of X, and r labelled directed edges. There is a directed edge labelled by δ from point X_k to point X_l if and only if $(X_k, X_l) \in H_\delta$. Let $r_\delta = |H_\delta|$. Then $r = \sum_{\delta\in\Delta} r_\delta$.

The values of n, m and r will be used to enumerate some simple graphs which will be considered in graph transformations.

A <u>transformation</u>, Q, is a graph replacement rule, denoted $L_1 =:: L_2$, where L_1 and L_2 are labelled graphs involving variables on the same set.

For example we define a transformation Q2 by the rule:

$L_1 =:: L_2$ where

$$L_1 = (\{a,b,c\},\ H_1 = \{(b,a)\},\ H_2 = \{(b,c)\})$$

$$L_2 = (\{a \cup b,\ c\},\ H_2 = \{(a \cup b,\ c)\})$$

Q2 is represented by the diagram:

Q2:
$$\begin{tikzcd} a & b \arrow[l, "1"'] \arrow[d, "2"] & =:: & a \cup b \arrow[d, "2"] \\ & c & & c \end{tikzcd}$$

A labelled graph, L, or a replacement rule, Q, is a logical expression about a system of relations on a set which may contain relations, variables, and operations between variables. Analogous to [1], we assume that the sets are lattices and that inverse and dual operators are defined on the relational system and on statements about the system.

In this case, an inverse operator is defined for each δ in Δ, and, since each relation is defined on the same set, there is only one dual operator, i.e. $D_L = D_R = D$. The group G of operators is generated by composition of inverse and dual operators defined in Table 1; where T is an operator in G and A is a logical expression such as a graph or transformation.

T	T (A)
E	A
D	A with lattice operations replaced by their dual operations
I_δ	A with H_δ replaced by H_δ^{-1}

Table 1

The group G is a commutative group with 2^{m+1} elements obtained from m different options of identity and inverse operators and the dual operator. The following notation will sometimes be used for composition of inverse operators:

$$I \text{ denotes } \prod_{\delta \varepsilon \Delta} I_\delta$$

$$I_{\delta_1 \delta_2} \text{ denotes } I_{\delta_1} I_{\delta_2}$$

In this paper, we will restrict our attention to the case when $m = 2$ and $\Delta = \{1, 2\}$, where the eight operators in G are:

$$E,\ I_1,\ I_2,\ I,\ D,\ I_1D,\ I_2D,\ ID.$$

Here $I = I_1 I_2$.

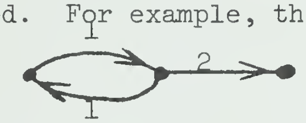
3. ENUMERATION OF SIMPLE LABELLED GRAPHS

In order to choose some representative labelled graphs as operands in transformations we will enumerate graphs with up to four points, two relations, and four edges, i.e. $n \leq 4$, $m \leq 2$, $r \leq 4$:

(i) Graphs which are isomorphic, i.e. identical except for a renaming of labels and points, are considered equivalent in the enumeration.

(ii) Graphs with reflexive loops will not be considered. Thus, edges will be exhibited between distinct points only and $n \geq 2$.

(iii) Multiple relations between two points will be treated as defining a new relation. Accordingly, graphs with multiple edges will not be enumerated. For example, the graph:



with n, m, r = 3, 2, 3 will be considered equivalent in the enumeration to the graph:

with n, m, r = 3, 2, 2 since $H_{1'}$ may be defined by $H_{1'} = H_1 \cap H_1^{-1}$

(iv) Disconnected graphs are treated by considering their connected components only and are omitted in the enumeration.

Figure 1 displays the enumerated graphs. Each row contains a graph, L, and all the inverse graphs I_1 (L), I_2 (L), I(L). Isomorphisms may exist between graphs in the same row only. Isomorphic graphs in a row are listed on the same line to the right of the row.

Except for the 2,1,1 case, graphs with m = 1 are not listed since they are special cases of graphs with m = 2 which are listed. For n = 2 or 3 and m = 2, a graph with any possible value of r is equivalent to one of the listed graphs. For n = 4 and m = 2, the 4,2,1 and 4,2,2 graphs are disconnected. All 4,2,3 graphs are listed.

Additional 4,2,4 graphs may be obtained by adding an edge to any corner of the 3,2,3 graphs. These graphs are not listed since transformations will be applied to their 3,2,3 subgraphs. Similarly, other more complex graphs will be handled by a series of transformations on simpler subgraphs.

FIGURE 1
SHEET 1

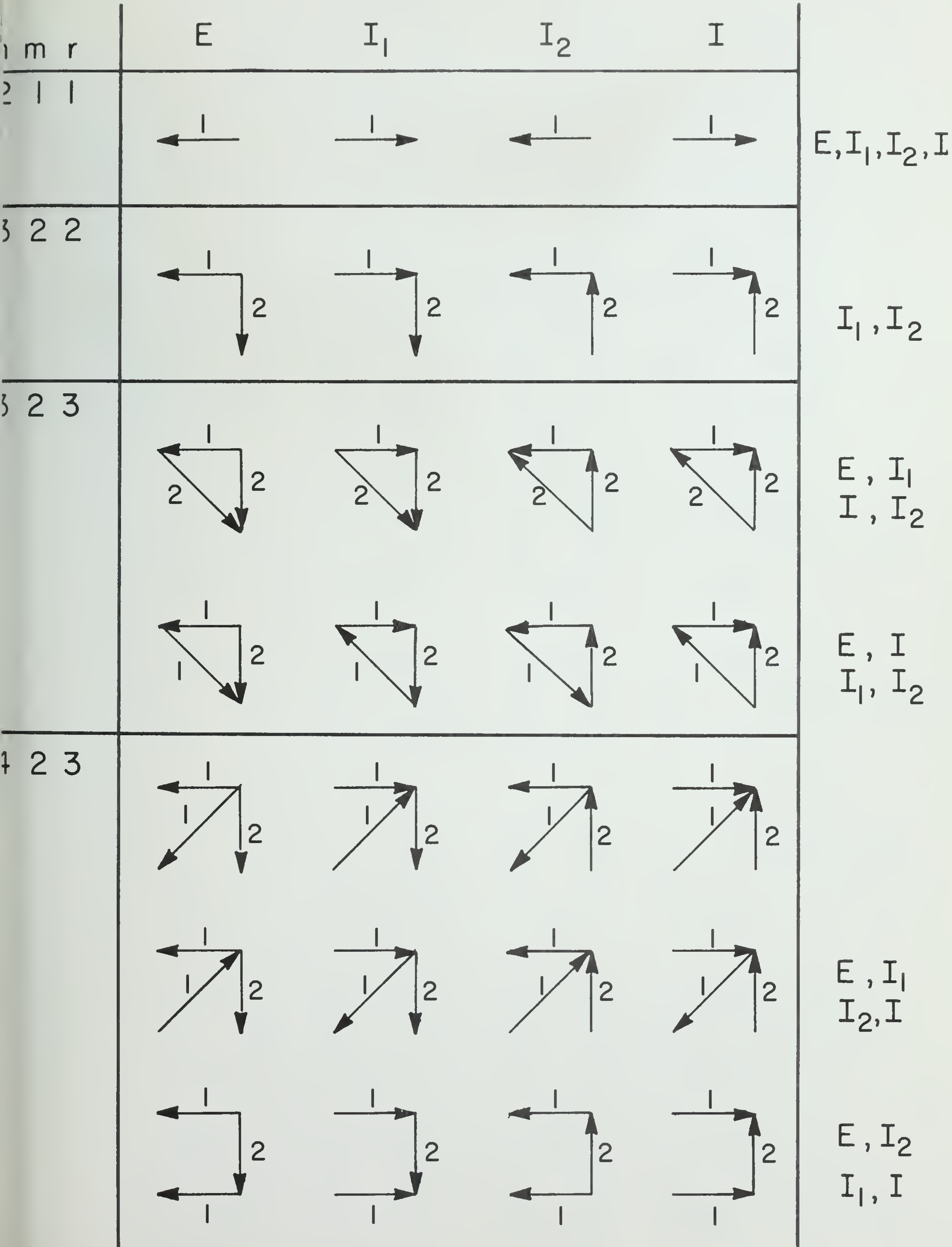
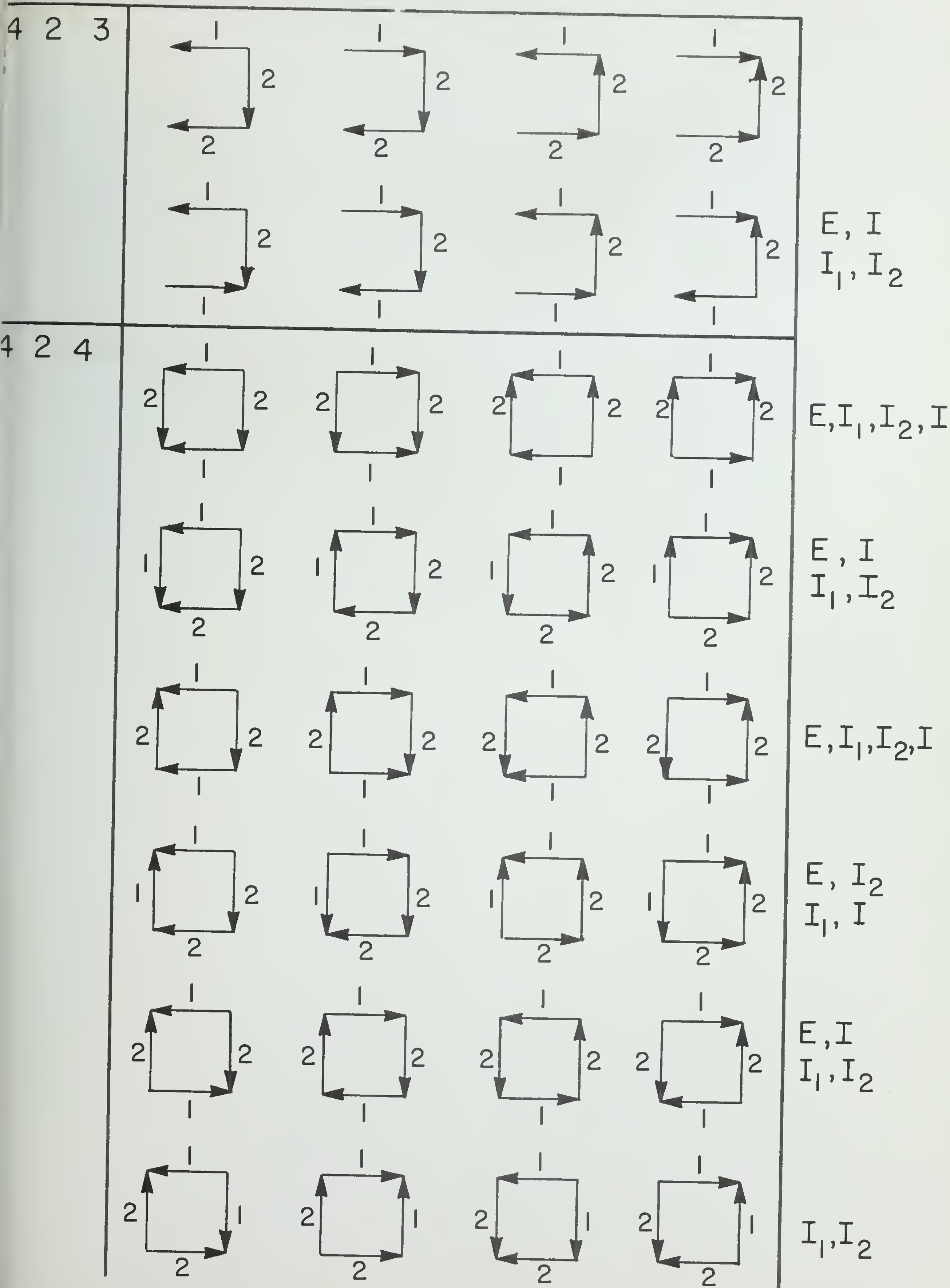


FIGURE 1
SHEET 2

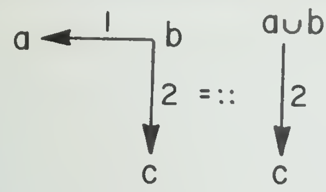
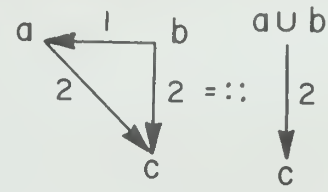
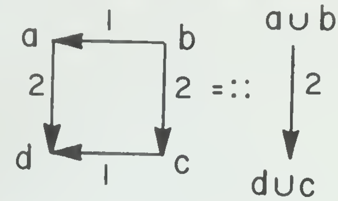
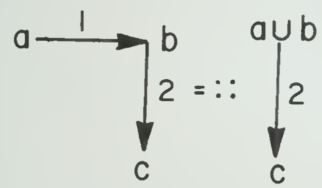
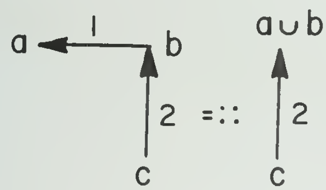
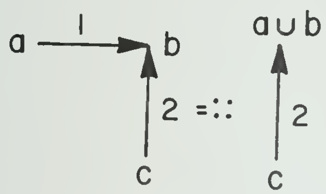
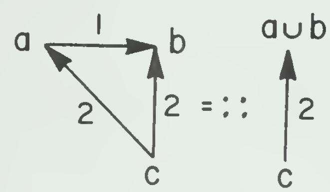
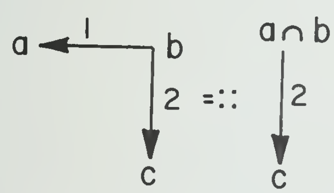
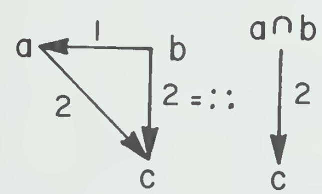
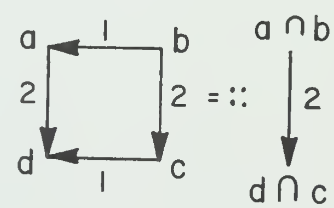
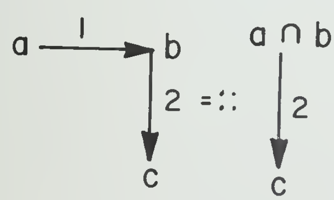
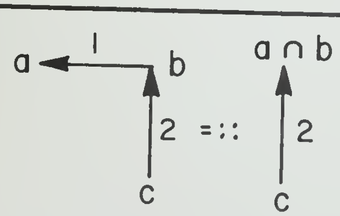
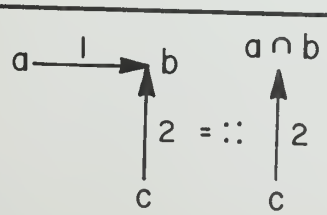
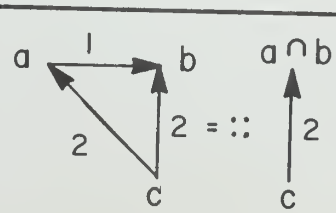


4. DEFINITION OF REPRESENTATIVE TRANSFORMATIONS

We choose three representative composite forming transformations which use graphs in Figure 1 as operands. One graph is selected for each r = 2,3,4. Figure 2 displays the three transformations, Q2, Q3, Q4 and also T(Q2), T(Q3), T(Q4) for the eight T in G. There are a total of fourteen non-isomorphic transformations generated by Q2, Q3, and Q4. The transformations form composites by taking the meet or the join of elements on the lefthand side of the transformation.

The basic criteria for the choice of transformations were simplicity and plausibility of composite formation. It is assumed that simple transformations, such as Q2, will occur most frequently and will be basic to any application procedure. More complex transformations are selected if one of the ways in which composites could be chosen appears intuitively most reasonable. For example, of the two basic 3,2,3 graphs in Figure 1, the one selected for Q3 gives an obvious choice of composite formation assuming that a relation can be extended to a composite if it holds with all the parts of the composite.

FIGURE 2 T(Q)

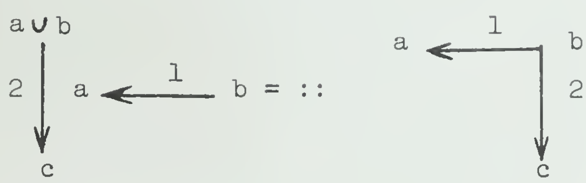
T \ Q	Q2	Q3	Q4
E			
I_1		Q3	Q4
I_2		I(Q3)	Q4
I			Q4
D			
I_1D		D(Q3)	D(Q4)
I_2D		ID(Q3)	D(Q4)
ID			D(Q4)

Reverse Transformations

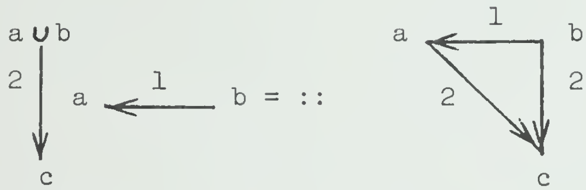
We will also consider the reverse transformations of the defined transformations. The reverse of a transformation Q will be denoted by $\hat{Q}$. $\hat{Q}$ is the transformation which restores all edges cut by the composite forming transformation Q, while assuming that the relations within the composite still exist.

$\widehat{Q2}$, $\widehat{Q3}$ and $\widehat{Q4}$ are shown below.

$\widehat{Q2}$:



$\widehat{Q3}$:



$\widehat{Q4}$:

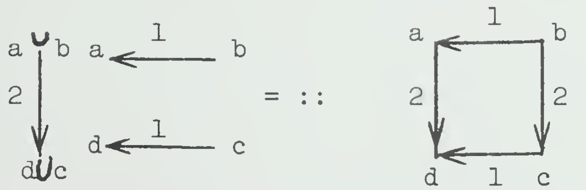
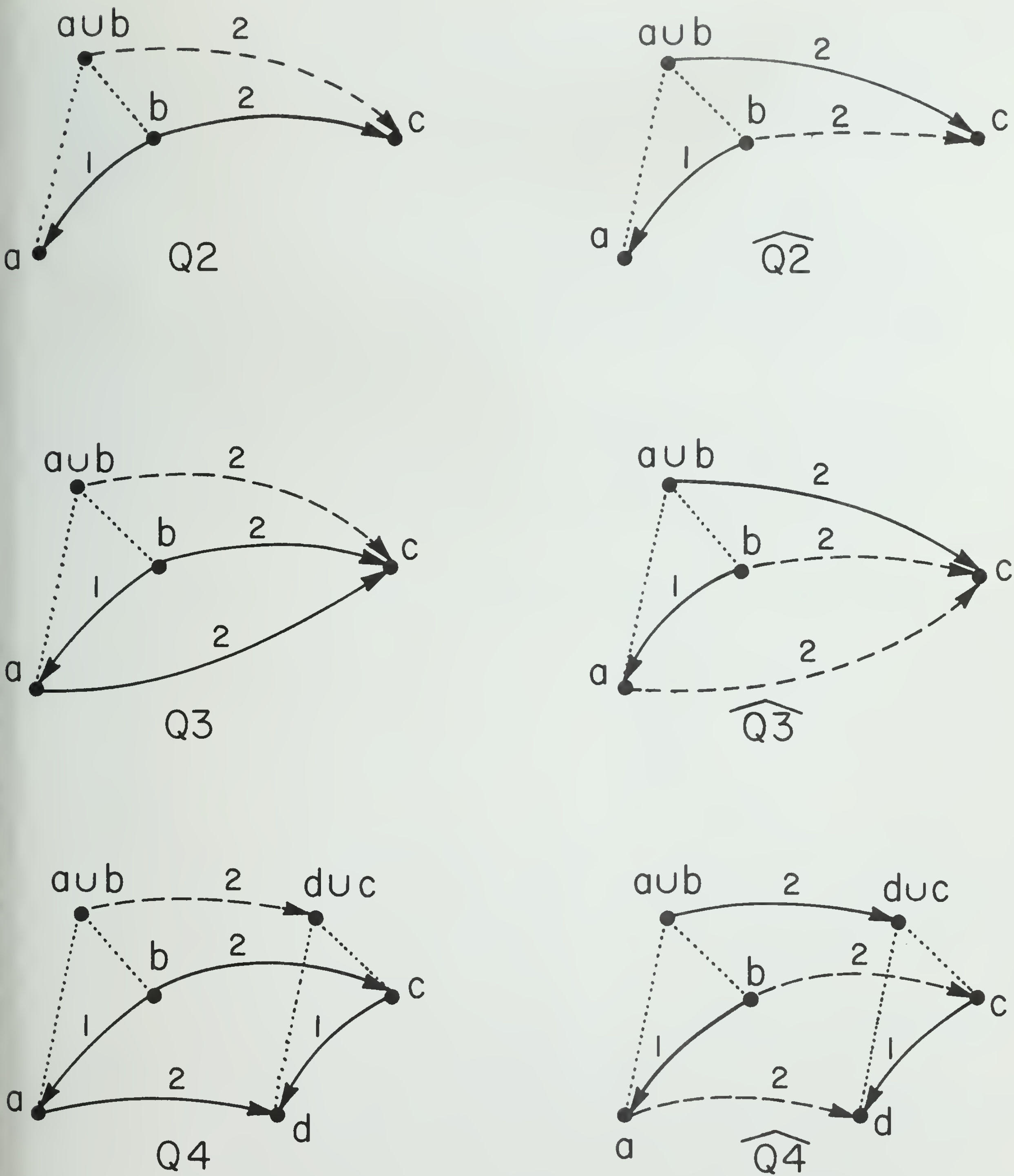


Figure 3 represents the transformations and reverse transformations by diagrams in which dotted straight lines indicate the lattice ordering between a composite element and its parts. Relations which are assumed by the transformation, i.e. those on the lefthand side, are indicated by unbroken curved edges and relations added are indicated by dashed curved edges.

FIGURE 3



5. SUFFICIENT CONDITIONS FOR TRANSFORMATIONS

In this section we will determine sufficient conditions such that the transformations defined earlier are always valid. A transformation, $Q = (L_1 =::L_2)$, is <u>valid</u> if and only if all relations contained in L_2 and not in L_1 are true.

Three different types of conditions are formulated:

1. Boolean expressions involving the composite formation properties of a single relation H, which were defined in [1]. These properties are denoted by P1, $P1^{-1}$, P2, $P2^{-1}$, P3, $P3^{-1}$, P4, $P4^{-1}$, P5, P6, P7, $P6^{-1}$.
2. The conditions that a single relation H is a subset of the defining relations of the lattice or their union or intersection. We denote the lattice containment relations by GTE, "greater than or equal," and LTE, "less than or equal."

Then the notation for the four conditions is given below.

Notation	Condition
GLTE	$H \subseteq GTE \cup LTE$
LTE	$H \subseteq LTE$
GTE	$H \subseteq GTE$
E	$H \subseteq GTE \cap LTE$

3. Boolean expressions involving other transformations Q. These conditions express implications between transformations and thus, unlike the previous two types of conditions, can depend on the properties of more than one relation.

The group of operators G is used to simplify the establishment of sufficient conditions for the transformations. As in [1], it is easily seen that any statement, A, involving relations, lattice operations, variables, and logical expressions is true if and only if T(A) is true, where T is an operator in G. That is, $A \Leftrightarrow T(A)$. In particular, we will use the fact:

$(A \Rightarrow Q) \Leftrightarrow (T(A) \Rightarrow T(Q))$

where A is a condition and Q is a transformation.

Table 2 lists alternate sufficient conditions for the fourteen non-isomorphic transformations T(Q2), T(Q3), T(Q4) and the fourteen reverse transformations $T(\widehat{Q2})$, $T(\widehat{Q3})$, $T(\widehat{Q4})$ for all T in G. All the conditions T(A) for T(Q) are determined from the conditions A for Q in the first row of Table 2. Each alternate condition for T(Q) is contained completely on one line within the row and column entry for T and Q unless the condition is enclosed in parenthesis. In Table 2 the conditions involving properties, P, (type 1 above) always apply to the relation H_2 of the transformation and the containment conditions (type 2 above) always apply to the relation H_1 of the transformation. For example, the meaning of the entry in row "E" and column "Q4" of Table 2 is expanded below:

(H_2 satisfies P5) or ($H_1 \subseteq$ GTE) or ($H_1 \subseteq$ LTE) or ((Q2 is valid) and (I_2(Q2) is valid)) $\Longrightarrow$ Q4 is valid.

The purpose in establishing these conditions is to be able to determine in as many cases as possible which transformations are valid by considering only abstract properties of the relations involved in the transformations. Then by characterizing all relations under consideration in terms of an adequate set of properties, a name-independent or context-free type of parse may be facilitated.

Table 2 - Sufficient Conditions for Transformations

<table>
<tr><th>Q
T</th><th>Q2</th><th>$\widehat{Q2}$</th><th>Q3</th><th>$\widehat{Q3}$</th><th>Q4</th><th>$\widehat{Q4}$</th></tr>
<tr><td>E</td><td>$P1^{-1}$
GTE</td><td>$P2^{-1}$
GTE
$\widehat{Q3}$
$I_1(\widehat{Q3})$</td><td rowspan="2">$P3^{-1}$
GLTE
Q2
$I_1(Q2)$</td><td rowspan="2">$P2^{-1}$
E
$(\widehat{Q2}$ &
$I_1(\widehat{Q2}))$</td><td rowspan="4">P5
GTE
LTE
(Q2 &
$I_2(Q2))$</td><td rowspan="4">P2 & $P2^{-1}$
E
$(\widehat{Q2}$ & $I_1(\widehat{Q2})$&
$I_2(\widehat{Q2})$&
$I(\widehat{Q2}))$</td></tr>
<tr><td>I_1</td><td>$P1^{-1}$
LTE</td><td>$P2^{-1}$
LTE
$I_1(\widehat{Q3})$
$\widehat{Q3}$</td></tr>
<tr><td>I_2</td><td>P1
GTE</td><td>P2
GTE
$I_2(\widehat{Q3})$
$I(\widehat{Q3})$</td><td rowspan="2">P3
GLTE
I(Q2)
$I_2(Q2)$</td><td rowspan="2">P2
E
$(I(\widehat{Q2})$&
$I_2(\widehat{Q2}))$</td></tr>
<tr><td>I</td><td>P1
LTE</td><td>P2
LTE
$I(\widehat{Q3})$
$I_2(\widehat{Q3})$</td></tr>
<tr><td>D</td><td>$P2^{-1}$
LTE</td><td>$P1^{-1}$
LTE
$D(\widehat{Q3})$
$I_1D(\widehat{Q3})$</td><td rowspan="2">$P4^{-1}$
GLTE
D(Q2)
$I_1D(Q2)$</td><td rowspan="2">$P1^{-1}$
E
$(D(\widehat{Q2})$&
$I_1D(\widehat{Q2}))$</td><td rowspan="4">P7
LTE
GTE
(D(Q2) &
$I_2D(Q2))$</td><td rowspan="4">P1 & $P1^{-1}$
E
$(D(\widehat{Q2})$ &
$I_1D(\widehat{Q2})$&
$I_2D(\widehat{Q2})$ &
$ID(\widehat{Q2}))$</td></tr>
<tr><td>I_1D</td><td>$P2^{-1}$
GTE</td><td>$P1^{-1}$
GTE
$I_1D(\widehat{Q3})$
$D(\widehat{Q3})$</td></tr>
<tr><td>I_2D</td><td>P2
LTE</td><td>P1
LTE
$I_2D(\widehat{Q3})$
$ID(\widehat{Q3})$</td><td rowspan="2">P4
GLTE
ID(Q2)
$I_2D(Q2)$</td><td rowspan="2">P1
E
$(ID(\widehat{Q2})$&
$I_2D(\widehat{Q2}))$</td></tr>
<tr><td>ID</td><td>P2
GTE</td><td>P1
GTE
$ID(\widehat{Q3})$
$I_2D(\widehat{Q3})$</td></tr>
</table>

6. TRANSFORMATION INDUCED CLASSIFICATION OF RELATIONS

Assume we have a set of transformations and a set of relations which may occur in the transformations. That is, the set H_Δ of relations in a transformation represents m variables which may be chosen from the class of relations under consideration. If we know the behavior (e.g. the validity) of the transformations for all possible combinations of relations, we can partition the relations into classes, such that all relations in the same class behave the same in a position in a given transformation with respect to all possible relations in the other positions in the transformation. Then by intersecting the partitions induced from each position in each transformation we will obtain a classification which is as fine-grained as necessary for the given relations and transformations. Thus, the only property of a relation now necessary for applying the transformations, as for example, in a name-independent parsing procedure, is the class to which the relation belongs.

The classification described above is an exhaustive technique which assumes complete information about all relations in all transformations and would be impractical for most situations. The approach taken in this section is to determine as much as possible about the transformation induced classification of relations on the basis of the properties defined previously. Although the exhaustive technique may be the only way to verify that the finest level classification has been obtained, a consideration of a priori properties will partially, and may completely, determine the classification in some cases.

Induced Partitions

First, we define and show the uniqueness of the transformation induced partitions. Given a set of relations, H, we consider a transformation, Q, involving m relations, H_δ, $\delta \in \Delta$, where each H_δ is a variable over the set H. Assume there is a function f_Q: $H^m \rightarrow N$ which associates some element of a set, N, with the transformation Q for any m relations in H.

Elements of N may indicate the validity or other properties of the transformation.

The function f_Q can be represented by an m-dimensional array, A, where the size of each dimension is the order of H. Assume $\Delta = \{1,2,\ldots, m\}$ and g and h are fixed elements of H. Then we can define m equivalence relations, $E_1, E_2, \ldots, E_m$, on H by:

$$g\ E_j h \iff f(H_1, H_2 \ldots, H_{j-1}, g, H_{j+1}, \ldots, H_m)$$
$$= f(H_1, H_2 \ldots, H_{j-1}, h, H_{j+1}, \ldots, H_m)$$

The m equivalence relations uniquely determine m partitions, $\Pi_1, \Pi_2, \ldots, \Pi_m$, on H. In terms of the array, A, the classes of Π_i correspond to equal hyperplanes perpendicular to the axis of the ith dimension. By combining equal hyperplanes, the array A can be reduced to an array where the size of the ith dimension is the number of classes in Π_i.

Thus, the partition Π_i uniquely classifies relations which are indistinguishable from each other in the ith position of the transformation for all relations in H.

Partial Classification

Now we consider the transformations Q2, Q3, Q4 and $\widehat{Q2}$, $\widehat{Q3}$, $\widehat{Q4}$. Here m = 2, $\Delta = \{1, 2\}$. Let $N = \{0, 1\}$ and define f_Q by:

$$f_Q(g,h) = \begin{cases} 1 & \text{If Q is always valid with } H_1 = g \text{ and } H_2 = h, \\ 0 & \text{Otherwise} \end{cases}$$

Then f_Q is represented by a matrix of binary values denoted by M_Q. Rows of M_Q correspond to relations H_1 and columns of M_Q correspond to relations H_2. The two partitions Π_1 and Π_2 group equal rows and columns respectively.

We partially determ ne the matrix M_Q by considering the conditions of Table 2. That is, we determine the number of different classes in Π_1 and Π_2 on the basis of all possible values of the conditions which imply H_1 and H_2. These results are summarized by Table 3 which shows the reduced matrix, M, for the six transformations Q2, $\widehat{Q2}$, Q3, $\widehat{Q3}$, Q4, and $\widehat{Q4}$. This matrix contains all the information of the individual matrices $M_{Q2}, \ldots, M_{\widehat{Q4}}$ and the partitions, Π_1 and Π_2, of M are the intersections of the partitions of the individual matrices. The five possible values of the containment properties for H_1 give five different classes in Π_1 (rows of M) and the twenty possible values of the properties P1, $P3^{-1}$, P5, P2, $P2^{-1}$ give fifteen different classes in Π_2 (columns of M). Since the matrix is only partially filled out, these are lower bounds on the total number of classes in Π_1 and Π_2.

We know also that the seven partially filled out matrices for the transformations T(Q2), T($\widehat{Q2}$),...,T($\widehat{Q4}$) for the other seven T in G, i.e. T $\neq$ E, will have values identical to M. In these matrices the conditions A are replaced by T(A) and the application of the dual or inverse operators to all conditions A does not change the possible values of the conditions.

Table 3

M

H_1 \ H_2	$P1^{-1}$	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	$P3^{-1}$	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
	P5	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0
	P2	1	0	x	1	0	x	1	0	x	1	0	x	1	0	x
	$P2^{-1}$	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
	Q2	1	1	1	1	1	1									
	$\widehat{Q2}$	1	1		1	1		1	1		1	1		1	1	
	Q3	1	1	1	1	1	1	1	1	1	1	1	1			
	$\widehat{Q3}$	1	1		1	1		1	1		1	1		1	1	
	Q4	1	1	1				1	1	1						
	$\widehat{Q4}$	1			1			1			1			1		
	Q2	1	1	1	1	1	1									
GTE	$\widehat{Q2}$	1	1		1	1		1	1		1	1		1	1	
$\cup$	Q3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
LTE	$\widehat{Q3}$	1	1		1	1		1	1		1	1		1	1	
	Q4	1	1	1				1	1	1						
	$\widehat{Q4}$	1			1			1			1			1		
	Q2	1	1	1	1	1	1									
	$\widehat{Q2}$	1	1		1	1		1	1		1	1		1	1	
LTE	Q3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q3}$	1	1		1	1		1	1		1	1		1	1	
	Q4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q4}$	1			1			1			1			1		
	Q2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q2}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
GTE	Q3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q3}$	1	1		1	1		1	1		1	1		1	1	
	Q4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q4}$	1			1			1			1			1		
	Q2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q2}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E	Q3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q3}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Q4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$\widehat{Q4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

7. EXAMPLE CLASSIFICATION

As an application of the ideas of the previous section, we will consider an example in which a labelled directed graph represents a simplified "house cartoon" of the type used in the paper by Ledley and Wilson (3).

The nodes of the initial graph correspond to the primitive elements, window, door, chimney, wall, roof and gable, of the cartoon. The set of relations, H, between nodes is given below.

<u>Relations</u>

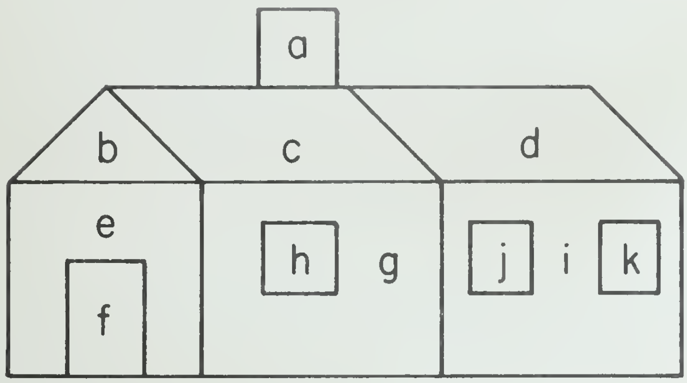
H_L	directly left of
H_T	directly on top of
H_C	contains
H_N	near

Figure 4 shows a house cartoon with labelled primitives, and the associated graph. Relations which are implied by other relations, such as "near" implied by "directly on top of", are not shown on the graph.

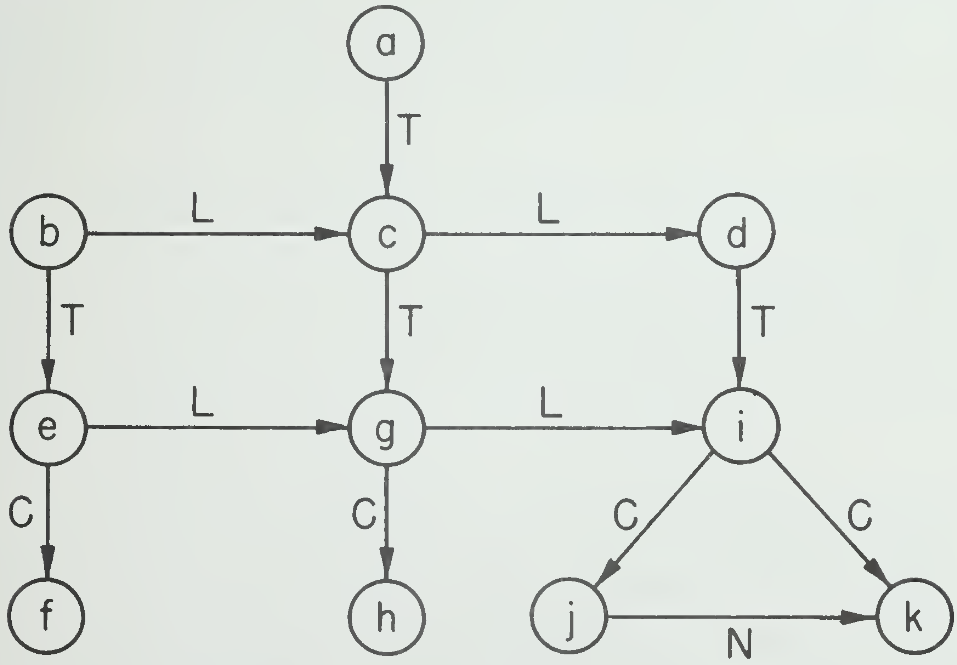
The goal now is to parse the graph by forming composite elements to obtain a final composite, "house", which can be represented by a linear string of relations and primitives.

The graph should be parsed so that no information from the original graph is lost. In considering the application of a composite forming transformation, Q, to the graph, we know that Q will be information lossless if both Q and the reverse of Q, $\widehat{Q}$, are always valid. In order to parse the graph, we would like to determine the validity of the transformations, Q2, $\widehat{Q2}$, Q3, $\widehat{Q3}$, Q4, $\widehat{Q4}$, for all combinations of the relations, H. Thus, we want to obtain the validity matrix, M, and a classification of the relations as described in Section 6.

We first partially determine the matrix on the basis of the abstract properties of the four relations. The properties used in M are shown in Table 4 for each relation H_δ.



HOUSE



PRIMITIVE GRAPH

Figure 4

δ \ Prop.	$P1^{-1}$	$P3^{-1}$	P5	P2	$P2^{-1}$	LTE	GTE
C	1	1	1	1	0	0	1
T	0	1	0	1	1	0	0
L	0	1	0	1	1	0	0
N	1	1	1	0	0	0	0

Table 4

From these properties we can use the entries in Table 3 of Section 6 directly, in order to partially fill in the matrix, M.

M is shown in Table 5.

By examining the matrix we can determine which transformations are always information lossless and thus can be carried out without any additional contextual or grammatical information.

Thus, Figure 4 is transformed into Figure 5 by applying only information lossless transformations Q2, I(Q2), and Q3 with $H_1 = H_C$, to form the composite elements $(e \cup f)$, $(g \cup h)$, $(j \cup k)$ and then $(i \cup j \cup k)$.

Depending on the exact definitions of the relations used, more information lossless transformations may be possible on the graph. These could be determined by examining specific transformations to complete the matrix M. All of these transformations depend only upon the type of relations involved in the transformations and not on the names of the objects or composites being formed.

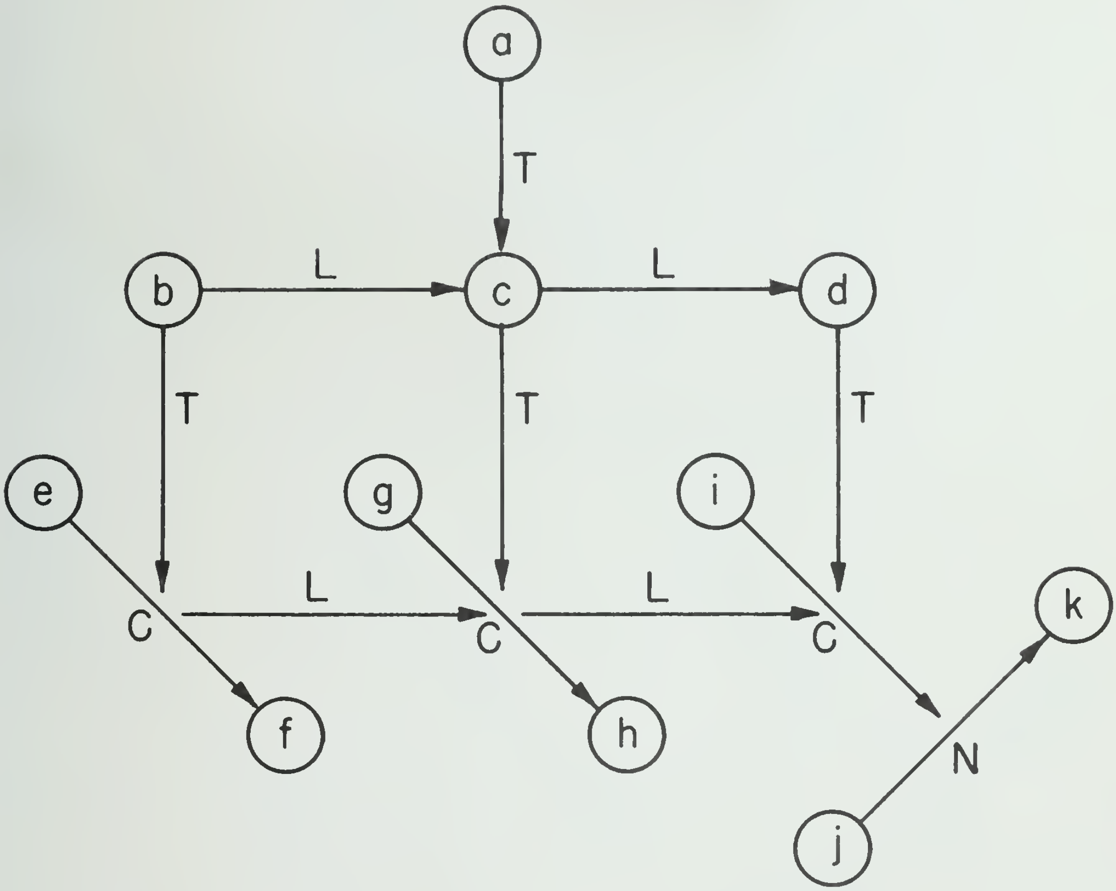
After no more information lossless transformations can be applied, the next level of parsing may have to be name dependent. For example, a model directed parse based on Ledley's house grammar, could now restrict the possible transformations so that the vertical composites are formed first. Under such a model, a series of Q2 and Q4 transformations would produce the graph shown in Figure 6, which is expressed in linear form below:

(bT(eCf))L((aTc)T(gCh))L(dT(iC(jNk)))

H_1 \ H_2	Q	C	T	L	N
C	Q2	1	1	1	1
	$\widehat{Q2}$	1	1	1	1
	Q3	1	1	1	1
	$\widehat{Q3}$		1	1	
	Q4	1	1	1	1
	$\widehat{Q4}$		1	1	
T	Q2	1			1
	$\widehat{Q2}$		1	1	
	Q3	1	1	1	1
	$\widehat{Q3}$		1	1	
	Q4	1			1
	$\widehat{Q4}$		1	1	
L	Q2	1			1
	$\widehat{Q2}$		1	1	
	Q3	1	1	1	1
	$\widehat{Q3}$		1	1	
	Q4	1			1
	$\widehat{Q4}$		1	1	
N	Q2	1			1
	$\widehat{Q2}$		1	1	
	Q3	1	1	1	1
	$\widehat{Q3}$		1	1	
	Q4	1			1
	$\widehat{Q4}$		1	1	

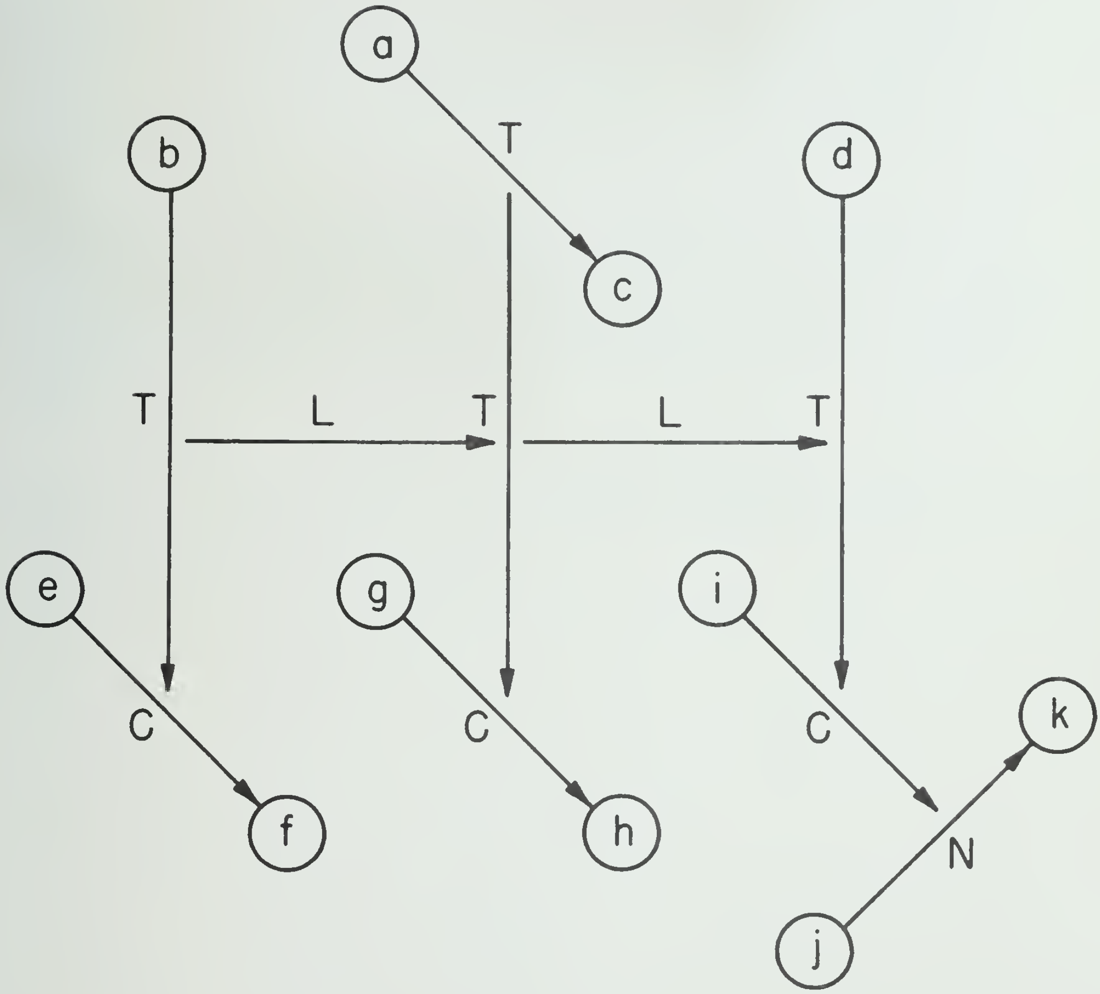
Table 5

Validity Matrix M For Transformations Q



PRIMITIVE GRAPH AFTER INFORMATION LOSSLESS TRANSFORMATIONS

Figure 5



GRAPH AFTER MODEL DIRECTED PARSING

Figure 6

References

1. Schwebel, J. C. and McCormick, B. H., Consistent Properties of Composite Formation Under a Binary Relation, University of Illinois, Department of Computer Science, Report No. 348, August 12, 1969.

2. McCormick, B. H., Syntax-Directed Recognition of Pictures: A New Theoretical Model, Talk presented at the conference: Emerging Concepts in Computer Graphics, University of Illinois, November 7, 1967.

3. Ledley, R. S. and Wilson, J. B., Concept Analysis by Syntax Processing, Proceedings ADI, pp. 1-8, 1964.

AEC-427
6/68)
:M 3201

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY–TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

EC REPORT NO.	2. TITLE
00-1018-1195	Graph Transformations for Composite Formation

YPE OF DOCUMENT (Check one):

- [x] a. Scientific and technical report
- [ ] b. Conference paper not to be published in a journal:
  - Title of conference ______
  - Date of conference ______
  - Exact location of conference ______
  - Sponsoring organization ______
- [ ] c. Other (Specify) ______

ECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

- [x] a. AEC's normal announcement and distribution procedures may be followed.
- [ ] b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
- [ ] c. Make no announcement or distrubution.

EASON FOR RECOMMENDED RESTRICTIONS:

UBMITTED BY: NAME AND POSITION (Please print or type)

John C. Schwebel

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61820

ignature	Date
John C. Schwebel	

FOR AEC USE ONLY

EC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION ECOMMENDATION:

ATENT CLEARANCE:

- [ ] a. AEC patent clearance has been granted by responsible AEC patent group.
- [ ] b. Report has been sent to responsible AEC patent group for clearance.
- [ ] c. Patent clearance not required.